

AD-A034 920

ARMY MISSILE RESEARCH DEVELOPMENT AND ENGINEERING LAB--ETC F/G 19/7  
FREE ROCKET SIX DEGREE OF FREEDOM DIGITAL SIMULATION.(U)  
NOV 76 M P STAMPS

UNCLASSIFIED

RF-77-1

NL

| OF |

AD  
A034920





ADA034920

(6)

TECHNICAL REPORT RF-77-1

**FREE ROCKET SIX DEGREE OF FREEDOM  
DIGITAL SIMULATION**

Mickle P. Stamps  
Advanced Systems Concepts Office  
US Army Missile Research, Development and Engineering Laboratory  
US Army Missile Command  
Redstone Arsenal, Alabama 35800

8 November 1976

Approved for public release; distribution unlimited

DDC  
RECEIVED  
JAN 27 1977  
C



**U.S. ARMY MISSILE COMMAND**  
Redstone Arsenal, Alabama 35800

**COPY AVAILABLE TO DDC DOES NOT  
PERMIT FULLY LEGIBLE PRODUCTION**



#### **DISPOSITION INSTRUCTIONS**

**DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED. DO NOT  
RETURN IT TO THE ORIGINATOR.**

#### **DISCLAIMER**

**THE FINDINGS IN THIS REPORT ARE NOT TO BE CONSTRUED AS AN  
OFFICIAL DEPARTMENT OF THE ARMY POSITION UNLESS SO DESIGNATED BY OTHER AUTHORIZED DOCUMENTS.**

#### **TRADE NAMES**

**USE OF TRADE NAMES OR MANUFACTURERS IN THIS REPORT DOES  
NOT CONSTITUTE AN OFFICIAL INDORSEMENT OR APPROVAL OF  
THE USE OF SUCH COMMERCIAL HARDWARE OR SOFTWARE.**



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RF-77-1	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) FREE ROCKET SIX DEGREE OF FREEDOM DIGITAL SIMULATION		5. TYPE OF REPORT & PERIOD COVERED Technical Report
7. AUTHOR(s) Mickie P. Stamps		6. PERFORMING ORG. REPORT NUMBER RF-77-1
9. PERFORMING ORGANIZATION NAME AND ADDRESS Commander US Army Missile Command ✓ Attn: DRSMI-RF Redstone Arsenal, Ala. 35809		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Commander US Army Missile Command Attn: DRSMI-RPR Redstone Arsenal, Ala. 35809		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS DA 1W362303A214 AMCMS 632303.214111.01
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE 8 November 1976
		13. NUMBER OF PAGES 32
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Free rocket simulation Six degrees of freedom Equations of motion Digital simulation		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes in detail a free rocket six degree of freedom trajectory program which utilizes a fourth order RUNGE-KUTTA subroutine. Several options such as programmed roll, directional control, and fold out fins are described. The program includes a listing of the equations, flow charts, program description, input and output information, and a complete program listing.		

## CONTENTS

	Page
I. INTRODUCTION. . . . .	3
II. PROGRAM DESCRIPTION . . . . .	3
III. PROGRAM LISTING . . . . .	25

ACCESSION for

MIS ☒ Warf Section

DOC ☐ Buff Section

UNANNOUNCED

JUSTIFICATION

BY \_\_\_\_\_

DISTRIBUTION/AVAILABILITY CODE

GEN. EVAL. NR. N. SPECIAL

A



## I. INTRODUCTION

The purpose of this report is to describe a six degree of freedom missile trajectory program in sufficient detail for the user to be able to run the program and/or make modifications to fit his specific needs. This program utilizes a nonrolling missile coordinate system to speed up run time when high (greater than 15 cps) roll rates occur. Several options are available such as directional control force, fold-out fins, and print-out of steady-state angles and velocity angular errors at boost burn out.

The program is in modular form. All variables to be integrated are stored in an array. A subroutine, DESUB, consists of the twelve simultaneous differential equations describing the behavior of these variables. A general RUNGE-KUTTA subroutine is utilized to integrate this set of equations. This report includes a listing of these equations, program description and flow charts, definitions of all symbols used, and a complete program listing.

## II. PROGRAM DESCRIPTION

### A. Equations of Motion

This program was written to simulate the following equations of motion to describe a free rocket trajectory from ignition to impact.

$$\dot{u} = (T - 1/2\rho S C_D u^2)/m + rv - qw - g \sin(QE + \theta)$$

$$\dot{v} = -ru + (1/2\rho S u^2 C_{N\alpha} [(v - w_v)/(u - w_u)] + T\delta_{tm} \cos(\phi + \phi_o) + k\psi)/m$$

$$\dot{w} = g \cos(QE + \theta) + qu + (-1/2\rho S u^2 C_{N\alpha} [(w - w_w)/(u - w_u)] + T\delta_{tm} \sin(\phi + \phi_o) - K\theta)/m$$

$$\dot{r} = (1/2\rho S d u^2 l_s C_{N\alpha} [(v - w_v)/(u - w_u)] - 1/4\rho S d^2 r u C_{mq} - T\delta_{tm} l_T d \cos(\phi + \phi_o) - k l_c d\psi + I_x p q)/I_y$$

$$\dot{q} = (1/2\rho S d u^2 l_s C_{N\alpha} [(w - w_w)/(u - w_u)] - 1/4\rho S d^2 q u C_{mq} + T\delta_{tm} d l_T \sin(\phi + \phi_o) - k l_c d\theta - I_x p r)/I_y$$

$$\dot{p} = (1/2\rho S d u^2 C_{l\delta} \delta_c - 1/4\rho S d^2 p u C_{lp} + 1/2\rho S d u^2 C_{l\delta} \delta_{10})/I_x + P_{spin}$$



$$\dot{x} = u \cos (QE + \theta) + w \sin (QE + \theta) \cos \psi - v \sin \psi$$

$$\dot{y} = u \cos (QE + \theta) \sin \psi + w \sin (QE + \theta) \sin \psi + v \cos \psi$$

$$\dot{z} = w \cos (QE + \theta) - u \sin (QE + \theta)$$

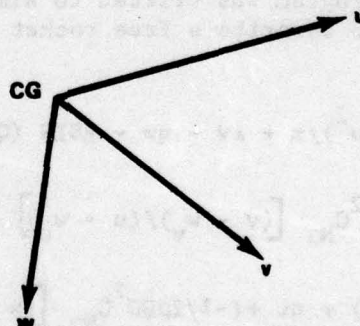
$$\dot{\theta} = q$$

$$\dot{\psi} = r / \cos (QE + \theta)$$

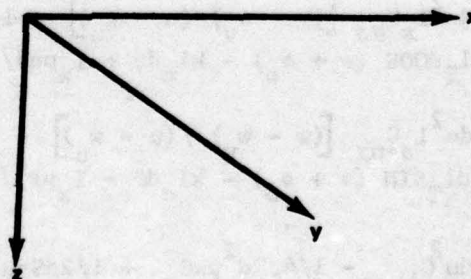
$$\dot{\phi} = p$$

If a constant roll rate or a programmed roll rate is to be implemented, this can be done by setting  $p = 0$  and  $\dot{\phi} = A_{spin}$ . This is noted in the program listing. In post flight analysis, frequently roll history can be acquired from telemetry data. This can be stored in tabular data in  $A_{spin}$  and utilized in the preceding manner.

These equations were converted to Fortran code by using the Fortran names listed in Table 1. These equations are calculated in the following coordinate systems:



Missile coordinate system.



Earth coordinate system.

## B. Program Outline

Solving the equations of motion requires current values of necessary atmospheric, aerodynamic, and missile physical characteristic data. The program is set up to read these data in by way of NAMELIST. This is one of the easiest methods of inputting data. An example of a data card for a constant is

$$x = 75.,$$

This is done with no regard for format except that only columns 2 through 72 may be used. If  $x$  is an array of five numbers, it is read as:

$$x(1) = 6., 4., 1., 17., 0.,$$

This can be continued onto the next card is necessary. All Fortran names used in the program are defined in Table 2. Program inputs are listed in Table 3.

Next, all variables to be integrated as a function of time are stored in an array, VARTB. VARTB(1) is time. The remainder of the array is ordered consistent with the order of the corresponding differential equations in DESUB.

The current values of atmospheric data are now calculated using the following equations:

$$G = G_0 \left[ 1 / (1 - z / 20855531.5) \right]^2$$

$$A = -z / (1 - z / 20855531.5)$$

$$\text{If } A \leq 36098.2388,$$

$$T = T_0 + (-0.00356616 A)$$

$$P = P_0 (T/T_0)^{0.1633654 G_0}$$

$$\rho = 28.966P / 49719.6257T$$

$$V_s = \sqrt{2403.075 T}$$

$$\text{If } A > 36089.2388,$$

$$T = 389.9878$$

$$P = 472.6765 e^{-0.00000149 G_0 (A - 36089.2388)}$$



$$\rho = 28.966 P/19390047.44$$

$$V_s = 968.075$$

where G = gravity

T = temperature

P = atmospheric pressure

$\rho$  = atmospheric density

$V_s$  = speed of sound.

The subscript 0 denotes the value of each of these variables at sea level.

Next, the wind components are transformed from the earth to missile coordinate system. These values are used to calculate missile total velocity and angles of attack.

Current values of missile physical characteristics and aerodynamic data are now calculated. The subroutine XINT is utilized for this purpose. This is simply a linear interpolation routine with one independent variable. Logic is employed in this section to determine if the missile is in boost or coast phase and whether fins have been unfolded.

Now the fourth-order RUNGE-KUTTA subroutine, SOLVDE/SSIMDE\* is called. The integration time step, DELT, is an input to this subroutine. A rule to follow in determining the size of the time step is not to allow any angle to change more than 7° to 8° during one step. This does not include the roll angle unless a thrust misalignment case is being run. The variable DELMIN is the minimum the program will allow DELT to be changed to. This is done in SOLVDE/SSIMDE. DELMIN can be set to the same size as DELT. Other inputs to this routine are VARTB, the table of variables to be integrated; WW, the dummy working array; IC; and N, the number of variables to be integrated. DESUB, the routine containing all the differential equations to be solved, is also required as an input argument. SOLVDE/SSIMDE is a two-purpose routine for integration. It has the capability of solving one nth order differential equation or n first order simultaneous differential equations.

After returning from the integration subroutine, several variables are calculated for output and printed if it is time for printing. If the missile altitude has become positive (below sea level) or if maximum

---

\*Sellers, William R., SOLVDE/SSIMDE, Directorate for Management Information Systems, US Army Missile Command.



simulation time has been reached, the program searches for the next case of data. Otherwise, it loops back to calculate atmospheric data and hence back through the program. Program output is defined in Table 4.

#### C. Subroutine DESUB

This subroutine contains all differential equations to be solved in the order that the variables are stored in VARTB in the main program. The subroutine inputs are V which corresponds to VARTB in MAIN, and J, the number of differential equations. A computed GO TO as a function of J controls the flow through the routine. For example, if  $J = 1$ , the first differential equation is solved and control is returned to the integration routine. The output is F which is the calculated value of the differential equation for the current value of J. Following is an example of a simple DESUB for solving three simultaneous differential equations,  $V(1) = x$ ,  $V(2) = y$ , and  $V(3) = z$ .

```
      SUBROUTINE DESUB (V, F, J)
      COMMON X,Y,Z
      DIMENSION V(4)
      GO TO (10, 20, 30), J
10    F = X
      RETURN
20    F = Y
      RETURN
30    F = Z
      RETURN
      END
```

TABLE 1. FORTRAN NAMES

Fortran Name	Symbol
VARTB (1)	time
VARTB (2)	w
VARTB (3)	u
VARTB (4)	v
VARTB (5)	r
VARTB (6)	q
VARTB (7)	x
VARTB (8)	y
VARTB (9)	z
VARTB (10)	$\theta$
VARTB (11)	$\psi$
VARTB (12)	p
VARTB (13)	$\phi$
XT	T
P	$\rho$
S	S
CD	$C_D$
XM	m
GRAV	g
QE	QE
CNA	$C_{N\alpha}$
WV	$w_v$
WU	$w_u$
WW	$w_w$
DTM	$\delta_{tm}$
XK	k
DX	d
SL	$l_s$
CMQ	$C_{mq}$

Table 1. (Concluded)

Fortran Name	Symbol
TL	$l_T$
CLA	$l_c$
ROLLI	$I_x$
XI	$I_y$
CLD	$C_{1\delta}$
DCANT	$\delta_c$
CLP	$C_{1p}$
CLO	$C_{10}$
PSPIN	$P_{spin}$



TABLE 2. SYMBOLS

Fortran Name	Definition
TEMPO	Temperature at sea level
PRSSO	Atmospheric pressure at sea level
GRAVO	Gravity at sea level
THETA	Flight path angle, pitch
IC	Initial condition for RUNGE-KUTTA routine
VARTB	Table of variables to be integrated
TT	Current value of time
WO	Initial velocity component of missile in pitch plane
U	Initial velocity component of missile along the longitudinal axis
VO	Initial velocity component of missile in yaw plane
RO	Initial yaw rate of missile
QO	Initial pitch rate of missile
PROLL	Initial roll rate of missile
XR	Initial range of missile in earth coordinate system
Y	Initial lateral displacement in earth coordinate system
Z	Initial vertical displacement in earth coordinate system
THETO	Initial pitch angle
PSIO	Initial yaw angle
PHIIN	Initial roll angle
WTIN	Initial missile total weight
WT	Current missile weight
XXTIN	Time of first print out
XXT	Time of next print out
KELL	Index indicating whether missile is on launcher
GRAV	Current value of gravity

TABLE 2, (Continued)

Fortran Name	Definition
TEMP	Current value temperature
PRSS	Current value of atmospheric pressure
P	Current value of atmospheric density
OM	Current value of speed of sound
WY	Boost phase lateral wind component
WY1	Coast phase lateral wind component
WX	Boost phase head wind component
WX1	Coast phase head wind component
WU	Component of head wind along longitudinal missile axis
WW	Component of head wind along vertical axis in missile coordinate system
VTOT	Total missile velocity
ALP	Angle of attack in vertical plane
ALPHA	Total angle of attack
UM	Current Mach number
ALTFT	Current altitude (positive up)
XT	Current missile thrust
T	Time table, independent variable for thrust table
A	Thrust table, $f(T)$
PSPIN	Roll acceleration due to spin motor
TTA	Time table, independent variable for PSPIN table
AR	PSPIN table, $f(TTA)$
TTA	Time table, independent variable for ASPIN table (Note both ASPIN and PSPIN will not be used in same simulation)
AK	ASPIN table, $f(TTA)$
ASPIN	Constant or programmed roll rate
CLP	Roll damping coefficient



TABLE 2. (Continued)

Fortran Name	Definition
UG	Mach number table, independent variable for CLP table
AO	CLP table, f(UG)
CLD	Roll moment coefficient due to fin deflection
UH	Mach number table, independent variable for CLD table
AP	CLD table, f(UH)
XK	Control force for system with direction control
TTB	Time table, independent variable for XK table
AL	XK table, f(TTB)
CMQ	Pitch damping coefficient
UE	Mach number table, independent variable for CMQ table
AN	CMQ table, f(UE)
CNA	Normal force coefficient
UB	Mach number table, independent variable for CNA table
AG	CNA table, f(UB), power on
CLO	Roll moment coefficient due to wraparound fin effect
UI	Mach number table, independent variable for CLO table
AS	CLO table, f(UI)
RNLM	Reynolds number per body length times Mach number
ZAL	Altitude table, independent variable for RNLM table
AU	RNLM table, f(ZAL)
BL	Missile body length (ft)
BLOD	Missile length divided by diameter (l/d)



TABLE 2. (Continued)

Fortran Name	Definition
RN	Reynolds number
CDF	Coefficient of drag due to skin friction
TB	Boost burn time
SPIMP	Specific impulse
TIMP	Total impulse
TPROP	Total propellant weight
DELT	Time integration step
XII	Initial transverse moment of inertia
XIB	Final transverse moment of inertia
TTO	Initial time for table values
XI	Current value of transverse moment of inertia
TLI	Initial thrust lever arm
TLF	Final thrust lever arm
TL	Current value of thrust lever arm
CLAI	Initial value of control lever arm
CLA	Current value of control lever arm
ROLLI	Current value of roll moment of inertia
ROLLA	Initial value of roll moment of inertia
ROLLB	Final value of roll moment of inertia
SL	Static margin
UA	Mach number table, independent variable for SL table
AI	SL table, power on, $f(UA)$
CD	Drag coefficient
UC	Mach number table independent variable for CD table, power on
AH	CD table, power on, $f(UC)$
UA	Mach number table, independent variable for SL table, power off
AJ	SL table, power off, $f(UA)$

TABLE 2. (Continued)

Fortran Name	Definition
UD	Mach number table, independent variable for CD table, power off
AM	CD table, power off, f(UD)
UJ	Mach number table, independent variable for CNA table, power off
AT	CNA table, power off, f(UJ)
XM	Missile mass
TFLD	Time of fin folding
DELMIN	Minimum time integration step
AX	Velocity along the x-axis in the earth coordinate system
AY	Velocity along the y-axis in the earth coordinate system
AZ	Velocity along the z-axis in the earth coordinate system
DRAG	Total missile drag
ACC	Total missile acceleration
THETS	Steady-state pitch angle after burn out
PSITS	Steady-state yaw angle after burn out
IBO	Index indicating whether to print out angular errors after burn
ZERR	Pitch angular error at burn out
YERR	Yaw angular error at burn out
XX	Number of time-steps per print out
TMAX	Maximum time - simulation stops after this time
LCS	Index indicating last case of data
QE	Quadrant elevation
S	Missile reference area
DTM	Thrust misalignment angle
PHIO	Initial thrust misalignment direction
DX	Missile diameter



TABLE 2. (Continued)

Fortran Name	Definition
ELL	Effective launcher length
DCANT	Fin cant angle
PCD	Percent effective drag coefficient-utilized in error budgets
J	Variable number to be integrated
V	Table of variables to be integrated-peculiar to subroutines DESUB and SOLVDE
W,WWW	Working arrays for integration routine
N	Number of differential equations
XX	Array of independent variables for interpolation purposes
YY	Array of dependent variables for interpolation purposes
X	Current value of independent variable
Y	Current value of dependent variable
TIME	Current value of time
U	Component of missile velocity along longitudinal axis
V	Component of missile velocity in yaw plane
W	Component of missile velocity in pitch plane
THET	Pitch angle
PSIO	Yaw angle
PHI	Roll angle
Q	Missile pitch rate
R	Missile yaw rate
ACC	Missile acceleration
VZ	Velocity along the z-axis in the earth coordinate system
VY	Velocity along the y-axis in the earth coordinate system
CNAI	Constant value of CNA before fins are unfolded



TABLE 2. (Concluded)

Fortran Name	Definition
SLI	Constant value of SL before fins are unfolded
VARTB (1)	Current value of time
VARTB (2)	Current value of W missile velocity component
VARTB (3)	Current value of U missile velocity component
VARTB (4)	Current value of V missile velocity component
VARTB (5)	Current value of R, missile yaw rate
VARTB (6)	Current value of Q, missile pitch rate
VARTB (7)	Current value of missile range
VARTB (8)	Current value of missile lateral displacement
VARTB (9)	Current value of missile altitude
VARTB (10)	Current value of missile pitch angle
VARTB (11)	Current value of missile yaw angle
VARTB (12)	Current value of missile roll rate
VARTB (13)	Current value of missile roll angle

TABLE 3. PROGRAM CONSTANT INPUTS

Fortran Name	Definition	Units
THETO	Initial pitch angle	rad
PSIO	Initial yaw angle	rad
PHIIN	Initial roll angle	rad
WTIN	Initial missile total weight	lb
TT	Current value of time	sec
WO	Initial velocity component of missile in pitch plane	ft/sec
U	Initial velocity component of missile along the longitudinal axis	ft/sec
VO	Initial velocity component of missile in yaw plane	ft/sec
RO	Initial yaw rate of missile	rad/sec
QO	Initial pitch rate of missile	rad/sec
PROLL	Initial roll rate of missile	rad/sec
XR	Initial range of missile in earth coordinate system	ft
Y	Initial lateral displacement in earth coordinate system	ft
Z	Initial vertical displacement in earth coordinate system	ft
XX	Number of time-steps per print out	-
XXTIN	Time of first print out	sec
TIMP	Total impulse	lb-sec
TPROP	Total propellant weight	lb
DELT	Time integration step	sec
XII	Initial transverse moment of inertia	slug-ft <sup>2</sup>
XIB	Final transverse moment of inertia	slug-ft <sup>2</sup>
TTO	Initial time for table values	sec
TLI	Initial thrust lever arm	cal
TLF	Final thrust lever arm	cal
WY	Boost phase lateral wind component	ft/sec
WY1	Coast phase lateral wind component	ft/sec



TABLE 3. (Continued)

Fortran Name	Definition	Units
WX	Boost phase head wind component	ft/sec
WX1	Coast phase head wind component	ft/sec
TFLD	Time of fin folding	sec
DELMIN	Minimum time integration step	sec
ROLLA	Initial value of roll moment of inertia	slug-ft <sup>2</sup>
ROLLB	Final value of roll moment of inertia	slug-ft <sup>2</sup>
CLAI	Initial value of control lever arm	cal
TB	Boost burn time	sec
IBO	Index indicating whether to print out angular errors after burn	0 - no print 1 - extra print
BL	Missile body length	ft
TMAX	Maximum time - simulation stops after this time	sec
LCS	Index indicating last case of data	0-not last case; 1-last case
QE	Quadrant elevation	rad
S	Missile reference area	ft <sup>2</sup>
DTM	Thrust misalignment angle	rad
PHIO	Initial thrust misalignment direction	rad
DX	Missile diameter	ft
ELL	Effective launcher length	ft
DCANT	Fin cant angle	rad
PCD	Percent effective drag coefficient utilized in error budgets	-
CNAI	Constant value of CNA before fins are unfolded	/rad
SLI	Constant value of SL before fins are unfolded	cal



TABLE 3. (Continued)

Fortran Name	Definition	Units
T	Time table, independent variable for thrust table	sec
A	Thrust table, f(T)	lb
TTA	Time table, independent variable for PSPIN table	sec
AR	PSPIN table, f(TTA)	rad/sec <sup>2</sup>
TTA	Time table, independant variable for ASPIN table (Both ASPIN and PSPIN will not be used in same simulation.)	sec
AK	ASPIN table, f(TTA)	rad/sec
UA	Mach number table independent variable for SL table, power off	-
AJ	SL table, power off, f(UA)	cal
UD	Mach number table, independent variable for CD table, power off	-
AM	CD table, power off, f(UD)	-
UJ	Mach number table, independent variable for CNA table, power off	-
AT	CNA table, power off, f(UJ)	/rad
UC	Mach number table, independent variable for CD table, power on	-
AH	CD table, power on, f(UC)	-
UI	Mach number table, independent variable for CLO table	-
AS	CLO table, f(UI)	-
UG	Mach number table, independent variable for CLP table	-
AO	CLP table, f(UG)	/rad
UB	Mach number table, independent variable for CNA table	-
AG	CNA table, f(UB), power on	/rad
UE	Mach number table, independent variable for CMQ table	-

TABLE 3. (Concluded)

Fortran Name	Definition	Units
AN	CMQ table, f(UE)	/rad
TTB	Time table, independent variable for XK table	sec
AL	XK table, f(TTB)	lb
UH	Mach number table, independent variable for CLD table	-
AP	CLD table, f(UH)	/rad
UA	Mach number table, independent variable for SL table	-
AI	SL table, power on, f(UA)	cal
ZAL	Altitude table independent variable for RNLM table	ft
AU	RNLM table, f(ZAL)	/ft

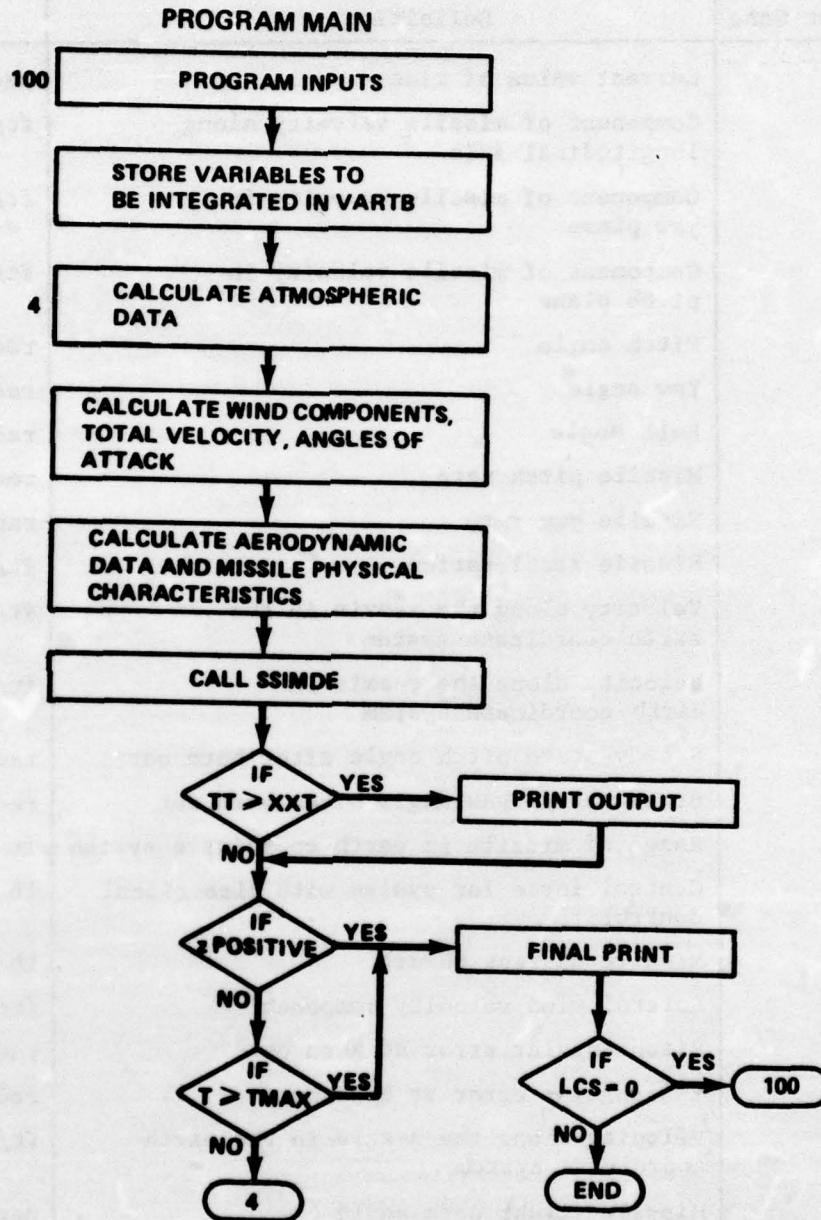


TABLE 4. PROGRAM OUTPUT

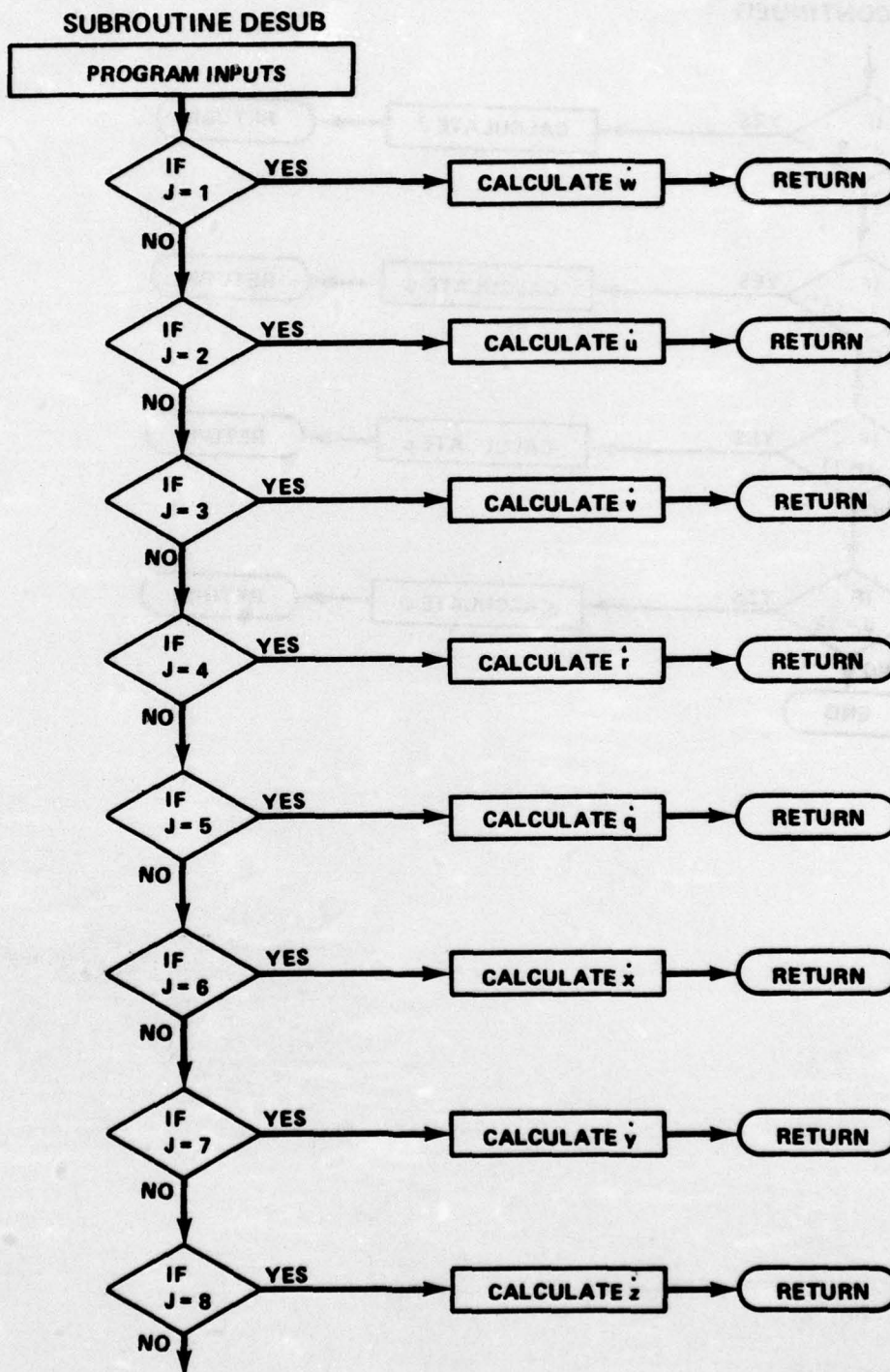
Fortran Name	Definition	Units
TIME	Current value of time	sec
U	Component of missile velocity along longitudinal axis	ft/sec
V	Component of missile velocity in yaw plane	ft/sec
W	Component of missile velocity in pitch plane	ft/sec
THET	Pitch angle	rad
PSIO	Yaw angle	rad
PHI	Roll angle	rad
Q	Missile pitch rate	rad/sec
R	Missile yaw rate	rad/sec
ACC	Missile acceleration	ft/sec <sup>2</sup>
VZ	Velocity along the z-axis in the earth coordinate system	ft/sec
VY	Velocity along the y-axis in earth coordinate system	ft/sec
THETS	Steady-state pitch angle after burn out	rad
PSITS	Steady-state yaw angle after burn out	rad
XR	Range of missile in earth coordinate system	ft
XK	Control force for system with directional control	lb
THRUST	Missile current thrust	lb
WY	Lateral wind velocity component	ft/sec
ZE	Pitch angular error at burn out	rad
YE	Yaw angular error at burn out	rad
XDOT	Velocity along the x-axis in the earth coordinate system	ft/sec
THA	Missile flight path angle	deg
ALP	Missile angle of attack in the vertical plane	rad



D. Flow Chart

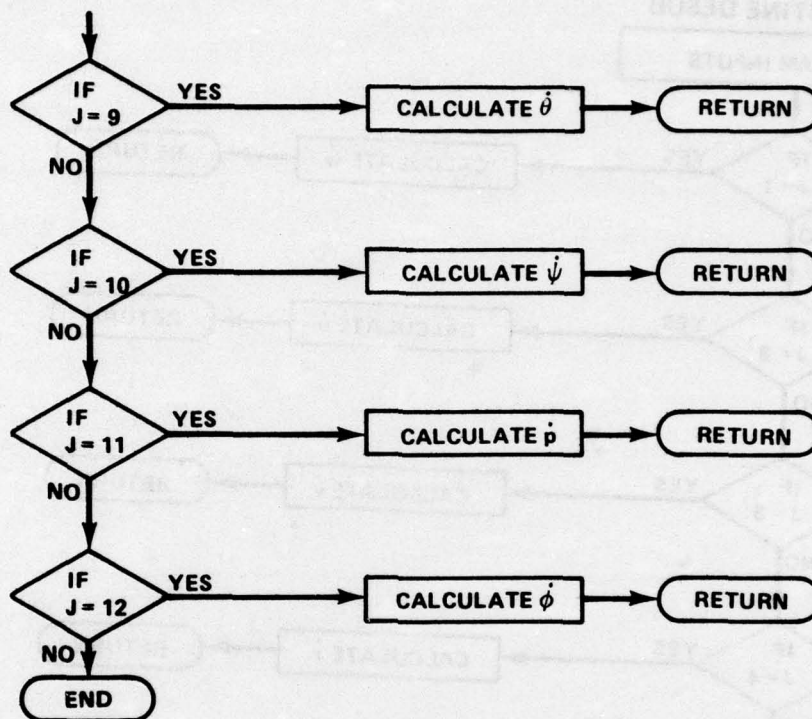


E. Flow Chart



F. Flow Chart

DESUB CONTINUED





### III. PROGRAM LISTING

```

PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
DIMENSION VARTH(13),WWW(3)
DIMENSION T(4),A(4),UA(13),AI(13),UR(13),AG(13),UC(8),AH(8),
*AJ(13),UG(2),AV(2),UH(2),AP(2),TTA(4),AR(4),AK(4),UT(7),AS(7),
*UJ(13),AT(13),ZAL(4),AU(4),TTH(2),AL(2),UCR(8),AM(8),UE(2),AN(2)
COMMON GRAV,UE,P,S,CNA,AM,XT,DTM,PHI,PHIO,XX,CD,WV,SL,XI,DX,CMU
COMMON TL,CL4,ROLLI,APHI,ELL,KELL,WW,WU,PSPIN,CLP,CLU,DCANT,PDOT
COMMON PCD,ASPIN,CLO
EXTERNAL DESUP
NAMELIST/NAMA/THETO,PHIIN,7,Y,TI,U,XR,WTIA,QO,W0,R0,V0,PSIO,
1 XXTIN,DTM,WY,ELL,UE,S,PHIO,UX,TLI,TLF,ROLLA,XX,TB,PROLL
2XII,XIB,TIMP,TPROP,TT0,CL4,DELT,TMAX,T,A,UA,AT,UR,AG,UC,AM,AJ,
3TTH,AL,UD,AM,UE,AN,DELMIN,POLLH,INO,WX,LCS,DCANT,UG,AO,UH,AP,
4TTA,AM,WYI,WX1,PCD,AK,UI,AS,UJ,AT,TFLO,ZAL,AU,HL,CNAI,SLI
RLOD=BL/UX
LCS=0
C
ATMOSPHERIC DATA AT SEALEVEL
TEMP0=518.688
PRESS0=2116.217
GRAV0=32.17404
GE=GRAV0*.1637653
100 READ(5,NAMA)
WRITE(6,1097)
1097 FORMAT(1H1)
WRITE(6,NAMA)
C
INITIAL CONDITIONS
THETA=GE*57.5
IC=0
VARTH(1)=TI
VARTH(2)=W0
VARTH(3)=U
VARTH(4)=V0
VARTH(5)=R0
VARTH(6)=I0
VARTH(7)=XX
VARTH(8)=Y
VARTH(9)=Z
VARTH(10)=THETO
VARTH(11)=PSIO
VARTH(12)=PROLL
VARTH(13)=PHIIN
XT=XTIN
XTT=XTIN
KELL=1
C
ATMOSPHERIC DATA CALCULATED
4 CAD=1.-VARTH(9)/2055531.5
GRAV=GRAV0*(1./CAD)**2
ALT=-VARTH(9)/UFI
IF (ALT-36.89.238H) 45,45,4H
45 TEMP=TEMP0*(1.-.00356616*ALT)
PRESS=PRESS0*(TEMP/TEMP0)**.5
P=28.966J*PRESS/(149719.6257*TEMP)
OM=(TEMP*2403.075)**.5
GO TO 70
46 TCD=389.9H7H
LSS=472.765*XM*(GRAV0*(-.00000149)*(ALT-3689.238H))
P=28.966J*PRESS/19390.47.44
U=55H.074
50 V=VY
IF (TI.GT.1H) V=VY1
C
AERODYNAMIC DATA CALCULATED
IF (TI.GT.TB) WX=XX1
U=WX*COS(VARTH(10)+UF)

```

```

WH=BX*SIN(VARTH(10)+UF)
VTOT=SQRT((VARTB(2)-WL)**2+(VARTH(4)-WY)**2+(VARTH(2)-WZ)**2)
ALP=(VARTH(2)-WH)/VTOT
ALPHA=SQRT((VARTH(2)-WH)**2+(VARTH(4)-WY)**2)/VTOT
UM=VTOT/OM
ALTFT=-VARTH(9)
CALL XINT(T,A,TT,AT)
CALL XINT(TTA,A-,TT,ASPI)
CALL XINT(TTA,AK,TT,ASPI)
CALL XINT(UG,AO,UM,CLP)
CALL XINT(UH,AP,UM,CLD)
CALL XINT(TTB,AL,TT,AK)
CALL XINT(HE,AK,UM,CMQ)
CALL XINT(UH,AG,UM,CNA)
CALL XINT(UJ,AS,UM,CLO)
CALL XINT(ZAL,AU,ALTFT,RNL)
RN=RNL*UM*HL
CDF=.294*HLOD*(1.0392-.1051*UM+.0035*UM**2)/(PN**2)
IF(TT.GT.TH) GO TO 101
C PHYSICAL CHARACTERISTICS UPDATED
6 SPIMP=TIMP/TPROP
WTDOT=XI/SPIMP
BT=BT-WTDOT*DELT
HBB=(XIB-XII)/(TH-TT0)
XI=XII+HBB*TT
HBT=(TLF-TLI)/(TB-TT0)
TL=TLI+HBT*TT
CLA=ABS(CLA1-TL)
ROLLI=ROLLA*((TT-TT0)/(TH-TT0))*(ROLLR-ROLLA)
CALL XINT(UA,AI,UM,SL)
CALL XINT(UC,AF,UM,CU)
GO TO 5
101 CALL XINT(UA,AJ,UM,SL)
CALL XINT(UO,AW,UM,CF)
CALL XINT(UJ,AT,UM,CNA)
5 XM=BT/GRAV
CD=CD+CDF
IF(TT.GT.TFLD) GO TO 7
CNA=CNA1
SL=SL1
7 CONTINUE
C INTEGRATION SUBROUTINE
IF(VARTH(13).GE.6.2832)VARTH(13)=VARTH(13)-6.2832
IF(VARTH(13).LE.-6.2832)VARTH(13)=VARTH(13)+6.2832
CALL SSIMDE(VARTB,WWW,DELT,DELMIN,12,IC,DESUB)
TT=VARTB(1)
AX=VARTH(3)*COS(VARTH(10)+QE)+VARTH(2)*SIN(VARTH(10)+QE)*
COS(VARTH(11))-VARTH(4)*SIN(VARTH(11))
AY=VARTH(4)+VARTH(3)*SIN(VARTH(11))
AZ=VARTH(2)*COS(VARTH(10)+QE)-VARTH(3)*SIN(VARTH(10)+QE)
THETA=ATAN((-AZ)/AX)*57.3
DWAQ=.5*P*SIN(VARTH(3)**2)
ACC=(XT-DWAQ)/XM
IF(TT.LT.TH) GO TO 200
C STEADY STATE ANGLES
THETS=VARTH(10)-VARTH(2)/VARTH(3)-VARTH(6)*XI/(DX*XM+VARTH(3)*SL)+
1 (VARTH(2)-WZ)*P*SIN(CMQ)/(4.*XM+VARTH(3)*SL)
PSITS=VARTH(11)+VARTH(4)/(VARTH(3)*COS(QE)+VARTH(2)*SIN(QE))-
1 VARTH(5)*XI/(DX*XM+VARTH(3)*SL*COS(QE))-(VARTH(4)-WY)*P*SIN(CMQ)/
2 (4.*XM+VARTH(3)*SL*COS(QE))
IF(TT.GF.XXT)WRITE(6,1049)THEIS,PSITS
200 IF(IMO.EQ.0) GO TO 201
C ANGULAR ERRORS AT HOUST MUEN OUT
VZ=VARTH(2)*COS(VARTH(10))-VARTH(3)*SIN(VARTH(10))
VY=VARTH(3)*COS(VARTH(10))+SIN(VARTH(11))+VARTH(4)*COS(VARTH(11))
1+VARTH(2)*SIN(VARTH(10))*SIN(VARTH(11))

```



```

ZERR=VZ/VTOT
YERR=VY/VTOT
IF (TT.GE.XXT)WRITE (6,1094)VZ,VY,ZERR,YERR
201 IF (TT-XXT)91,90,90
90 WRITE (6,1090)VARTH(1),VARTH(3),VARTH(7),XW
WRITE (6,1091)VARTH(10),VARTH(2),AZ,VARTH(9)
WRITE (6,1092)VARTH(11),VARTH(4),AY,VARTH(8)
WRITE (6,1093)VARTH(13),VARTH(6),VARTH(5),URAG
WRITE (6,1095)XT,VY,XK,ACC
WRITE (6,1100)X,THETA,ALP
WRITE (6,1096)
1090 FORMAT (8M TIME= ,E14.7,2X,4MU= ,E14.7,2X,4MXR= ,E14.7,2X,
17MMASS= ,E14.7)
10910 FORMAT (8M THET= ,E14.7,2X,4MU= ,E14.7,2X,4MAZ= ,E14.7,2X,
17M2= ,E14.7)
10920 FORMAT (8M PSIO= ,E14.7,2X,4MV= ,E14.7,2X,4MAY= ,E14.7,2X,
17MV= ,E14.7)
10930 FORMAT (8M PHI= ,E14.7,2X,4MQ= ,E14.7,2X,4MR= ,E14.7,2X,
17MDRAG= ,E14.7)
1095 FORMAT (8M THRUST= ,E14.7,2X,4MUY= ,E14.7,2X,4MXK= ,E14.7,2X,4HACC=,
1E14.7)
1096 FORMAT (1M )
1098 FORMAT (8M VZ= ,E14.7,2X,4MVY= ,E14.7,2X,4MZE= ,E14.7,2X,4MYE= ,
1E14.7)
1099 FORMAT (8M THEYS= ,E14.7,2X,4MPTS=,E14.7)
1100 FORMAT (8M XDOT= ,E14.7,2X,4MTHA=,E14.7,2X,4MALP=,E14.7)
XXT=XXT+XX*DELT
91 CONTINUE
C PROGRAM STOPS ON NEGATIVE ALTITUDE OR MAX TIME
IF (VARTH(9).GT.10.)GO TO 300
IF (TT-TMAX)4,300,300
300 WRITE (6,1090)VARTH(1),VARTH(3),VARTH(7),XW
WRITE (6,1091)VARTH(10),VARTH(2),AZ,VARTH(9)
WRITE (6,1092)VARTH(11),VARTH(4),AY,VARTH(8)
WRITE (6,1093)VARTH(13),VARTH(6),VARTH(5),DRAG
WRITE (6,1095)XT,VY,XK,AU
WRITE (6,1094)VZ,VY,ZERR,YERR
C LAST SET OF DATA SIGNALLED BY LCS=1
301 IF (LCS.EQ.0)GO TO 100
CALL EXIT
END
SUBROUTINE DESUP (V,F,J)
COMMON GRAV,GE,P,S,CNA,XK,XT,DTM,PHI,PHIO,XK,CD,MV,SL,XI,DX,CMD
COMMON TL,CLA,NOLLI,APHI,ELL,KELL,MW,WU,PSPIN,CLP,CLD,OCANT,PDOT
COMMON PCD,ASPIA,CLO
DIMENSION V(13)
IF (V(7).GT.ELL)KELL=?
GO TO (10,20,30,40,50,60,70,80,90,100,110,120),J
10 F=0
GO TO (12,11),KELL
11 F=GRAV*COS(V(10)+QE)+V(5)*V(3)-(V(3)**2)*P*S*CNA*(V(2)-WU)/
1(2.*XK*(V(3)-WU))+
2(XT*DTM*SIN(V(13)+PHIO)+XK*V(11))/XK
12 RETURN
20 F=(XT-.5*P*S*PCD*CU*V(3)**2)/XK+V(4)*V(5)-V(2)*V(6)
GO TO (22,21),KELL
21 F=(XT-.5*P*S*PLC*CU*V(3)**2)/XK-GRAV*SIN(V(10)+QE)+V(4)*V(5)-V(2)*
1V(6)
22 RETURN
30 F=0
GO TO (32,31),KELL
31 F=(-1.*V(5)*V(3))-(V(4)-V(6))*(V(3)**2)*P*S*CNA/(2.*XK*(V(3)-WU))+
1(XT*DTM*COS(V(13)+PHIO)+XK*V(11))/XK
32 RETURN
40 F=0
GO TO (42,41),KELL

```



```

41 F=(V(3)**2)*P*S*D*X*CN*(V(4)-W)*SL/(2.*X*(V(3)-W))-
1 .5*(DX**2)*P*S*V(5)*V(3)*
2 CMQ/(2.*X1)+(-1.*XT*DT*COS(V(13)+PHI0)*TL*DX-1.*DX*CLA*XX*
3V(11))/X1+ROLLI*V(12)*V(6)/X1
42 RETURN
50 F=0
GO TO (52,51),KLL
51 F=(-1.*(V(3)**2)*P*S*D*X*CN*(V(2)-W)*SL)/(2.*X*(V(3)-W))-
1(.5*P*S*(DX**2)*V(6)*
2V(3)*CMQ)/(2.*XT)+(DX*TL*XT*DT*SIN(V(13)+PHI0)-DX*XX*V(15)*CLA)/
3XI-ROLLI*V(12)*V(5)/X1
52 RETURN
60 F=V(3)*COS(V(10)+QE)+V(2)*SIN(V(10)+QE)*COS(V(11))-V(4)*SIN(V(11))
RETURN
70 F=0
GO TO (72,71),KELL
71 F=V(3)*COS(V(10)+QE)*SIN(V(11))+V(2)*SIN(V(10)+QE)*SIN(V(11))+
1V(4)*COS(V(11))
72 RETURN
80 F=V(2)*COS(V(10)+QE)-V(3)*SIN(V(10)+QE)
RETURN
90 F=0
GO TO (92,91),KFL
91 F=V(6)
92 RETURN
100 F=0
GO TO (102,101),KELL
101 F=V(5)/COS(V(10)+QE)
102 RETURN
110 F=-P*S*(DX**2)*V(3)*V(12)*CLP/(4.*ROLLI)*P*S*(V(3)**2)*DX*CLD*
1DCANT/(2.*ROLLI)+PSPIN+.5*P*S*DX*CLD*V(3)**2/ROLLI
F=0.
PUOT=F
RETURN
120 F=V(12)
F=ASPIN
RETURN
END
SUMROUTINE SOLVCE(V,W,DEL,DELMIN,N,IC,DESUB)

```

REMOVE THESE TWO CARDS IF THE ROLL  
RATE IS NOT CONSTANT OR PROGRAMMED

C			SVUE0000
C			SVUE0010
C			SVUE0020
C			SVUE0030
C			SVUE0040
C	1 V	VARIABLE TABLE IE.	SVUE0050
C		V(1) = X	SVUE0060
C		V(2) = Y	SVUE0070
C		V(3) = Y PRIME	SVUE0080
C		V(4) = Y TWO PRIME	SVUE0090
C		.....	SVUE0100
C		V(N+1) = Y (N-1) PRIME	SVUE0110
C		V(N+2) = D(N)Y/DX(N) UPON RETURN	SVUE0120
C	2 W	WORKING ARRAY. A DIFFERENT ARRAY MUST BE USED FOR	SVUE0130
C		EACH DIFFERENT EQUATION OR SYSTEM OF EQUATIONS BEING	SVUE0140
C		EVALUATED. THE DIMENSION MUST BE AT LEAST W(3).	SVUE0150
C	3 DEL	STEP SIZE FOR INDEPENDENT VARIABLE. AN INCREASING-	SVUE0160
C		DECREASING VARIABLE STEP SIZE (H) IS USED INTERNALLY	SVUE0170
C		IN THE ROUTINE. RETURN TO THE CALLING PROGRAM WILL	SVUE0180
C		BE AT V(1)+DEL.	SVUE0190
C	4 DELMIN	MINIMUM VALUE TO BE USED FOR (H) WHEN DECREASING	SVUE0200
C		THE INTEGRATION INCREMENT.	SVUE0210
C	5 N	ORDER OF THE EQUATION TO BE SOLVED.	SVUE0220
C	6 IC	USED FOR INTERNAL CONTROL. THIS VARIABLE MUST BE	SVUE0230
C		SET EQUAL TO ZERO FOR THE FIRST INTEGRATION STEP.	SVUE0240
C		A DIFFERENT VARIABLE MUST BE USED FOR EACH DIFFERENT	SVUE0250
C		EQUATION OR SYSTEM OF EQUATIONS BEING EVALUATED. DO	SVUE0260
C		NOT USE A LITERAL SINCE IC IS CONTINUALLY CHANGED.	SVUE0270

```

C      7 DESUB NAME OF ROUTINE CONTAINING THE EQUATION TO BE SOLVED. SVDE0280
C      THE NAME USED IN THE CALLING PROGRAM MUST BE DEFINED SVDE0290
C      IN A EXTERNAL STATEMENT IN THE CALLING PROGRAM. SVDE0300
C
C      SUBROUTINE SSIMDE(V,W,DEL,DELMIN,N,IC,DESUB) SVDE0310
C      GIVEN THE FIRST ORDER SYSTEM OF DIFFERENTIAL EQUATIONS SVDE0320
C      DY1/DX = F1(X,Y1,...,YN) SVDE0330
C      DY2/DX = F2(X,Y1,...,YN) SVDE0340
C      ..... SVDE0350
C      DYN/DX = FN(X,Y1,...,YN) SVDE0360
C
C      1 N VARIABLE TABLE IE. SVDE0370
C      V(1) = X SVDE0380
C      V(2) = Y1 SVDE0390
C      V(3) = Y2 SVDE0400
C      ..... SVDE0410
C      V(N+1) = YN SVDE0420
C
C      2 W SAME AS 2 ABOVE. SVDE0430
C      3 DEL SAME AS 3 ABOVE. SVDE0440
C      4 DELMIN SAME AS 4 ABOVE. SVDE0450
C      5 N NUMBER OF SIMULTANEOUS EQUATIONS TO BE SOLVED. SVDE0460
C      6 IC SAME AS 6 ABOVE. SVDE0470
C      7 DESUB SAME AS 7 ABOVE. SVDE0480
C
C      THE VARIABLES C AND VS IN THIS PROGRAM MUST BE DIMENSIONED
C      AT LEAST C(4,N) AND VS(N+2). SVDE0490
C
C      10 APRIL 69 SVDE0500
C      RALPH SELLERS SVDE0510
C
C      DOUBLE PRECISION V,W,DEL,DELMIN,C,VS,TSTOP,H,M2,F,TMP1 SVDE0520
C
C      EXTERNAL DESUB SVDE0530
C      DIMENSION V(1),W(1),C(4,15),VS(17) SVDE0540
C
C      1SYS = 0 SVDE0550
C      NP1 = N+1 SVDE0560
C      NP2 = NP1+1 SVDE0570
C      NTOT = NP2 SVDE0580
C      GO TO 3 SVDE0590
C      ENTRY SSIMDE SVDE0600
C      1SYS = 1 SVDE0610
C      NP1 = N+1 SVDE0620
C      NP2 = NP1+1 SVDE0630
C      NTOT = NP1 SVDE0640
C
C      COMPUTE STOP FOR THIS INTEGRATION STEP SVDE0650
C      3 TSTOP = V(1) * DEL SVDE0660
C      CHECK FOR FIRST TIME INTO ROUTINE SVDE0670
C      10 IF (IC .GT. 0) GO TO 20 SVDE0680
C      FIRST TIME IN SVDE0690
C      13 M = DEL SVDE0700
C      NP = 0 SVDE0710
C      W(2) = DEL SVDE0720
C      GO TO 60 SVDE0730
C
C      RESTART IF INPUT DEL HAS CHANGED SVDE0740
C      20 IF (DEL .NE. W(2)) GO TO 13 SVDE0750
C      M = W(1) SVDE0760
C      W(2) = W(1) SVDE0770
C
C      TEST FOR DOUBLING M. INCREASE TEST AS NUMBER OF HALVINGS SVDE0780
C      INCREASES. SVDE0790
C      40 IF (IC .LT. 10*(NM+1)) GO TO 60 SVDE0800
C      IC = 1 SVDE0810
C
C      DOUBLE M SVDE0820
C      50 M2 = 2.*M SVDE0830
C      IF (V(1)+M2 .GT. TSTOP) GO TO 60 SVDE0840

```



```

      IF (M2 .LE. DEL) GO TO 53
      GO TO 60
53  H = M2
C    INCREASE COUNTER
60  IC = IC + 1
C    INTEGRATE USING R-K
C    SAVE V TABLE VALUES IN VS ARRAY
105 DO 106 I=1,NTOT
106 VS(I) = V(I)
C    FIRST PASS THRU R-K COMPUTATION
110 DO 114 J=1,N
      IF (ISYS .NE. 0) GO TO 115
      IF (J .EQ. N) GO TO 115
      F = V(J+2)
      GO TO 119
115 CALL DESUB(V,F,J)
119 C(1,J) = F * H
C    SECOND AND THIRD PASS THRU R-K CALCULATION
120 V(1) = VS(1) + .5*M
      DO 150 I=2,3
      DO 130 J=2,NP1
130 V(J) = VS(J) + .5*C(1-J-1)
      DO 140 J=1,N
      IF (ISYS .NE. 0) GO TO 135
      IF (J .EQ. N) GO TO 135
      F = V(J+2)
      GO TO 140
135 CALL DESUB(V,F,J)
140 C(1,J) = F * H
150 CONTINUE
      IF (.5*M .LT. DELMIN) GO TO 160
C    TEST FOR HALVING H
      DO 153 J=1,N
      TMP1 = (C(2,J)-C(3,J)) / (C(1,J)-C(2,J))
      IF (ABS(TMP1) .GT. .025) GO TO 155
153 CONTINUE
      GO TO 160
C    HALVE H
155 H = .5*M
      NP = NP/2
      IC = 1
      DO 157 J=1,NTOT
157 V(J) = VS(J)
      GO TO 110
C    FOURTH PASS THRU R-K CALCULATION
160 V(1) = VS(1) + F
      DO 161 J=2,NP1
161 V(J) = VS(J) + C(1,J-1)
      DO 170 J=1,N
      IF (ISYS .NE. 0) GO TO 165
      IF (J .EQ. N) GO TO 165
      F = V(J+2)
      GO TO 170
165 CALL DESUB(V,F,J)
170 C(4,J) = F * H
C    UPDATE V TABLE
      DO 180 J=1,N
180 V(J+1)=VS(J+1)+(C(1,J) + 2.000*(C(2,J)+C(3,J)) + C(4,J))/ 6.000
C    TEST FOR END OF INTEGRATION STEP
200 IF (V(1) + M/2.0 .GT. TSTOP) GO TO 220
      GO TO 40
C    END OF REQUIRED INTEGRATION
220 V(1) = TSTOP
      IF (ISYS .NE. 0) GO TO 905
      CALL DESUB(V,V(NP2),1)
999 CONTINUE

```

```

SVDE0940
SVDE0950
SVDE0960
SVDE0970
SVDE0980
SVDE0990
SVDE1000
SVDE1010
SVDE1020
SVDE1030
SVDE1040
SVDE1050
SVDE1060
SVDE1070
SVDE1080
SVDE1090
SVDE1100
SVDE1110
SVDE1120
SVDE1130
SVDE1140
SVDE1150
SVDE1160
SVDE1170
SVDE1180
SVDE1190
SVDE1200
SVDE1210
SVDE1220
SVDE1230
SVDE1240
SVDE1250
SVDE1260
SVDE1270
SVDE1280
SVDE1290
SVDE1300
SVDE1310
SVDE1320
SVDE1330
SVDE1340
SVDE1350
SVDE1360
SVDE1370
SVDE1380
SVDE1390
SVDE1400
SVDE1410
SVDE1420
SVDE1430
SVDE1440
SVDE1450
SVDE1460
SVDE1470
SVDE1480
SVDE1490
SVDE1500
SVDE1510
SVDE1520
SVDE1530
SVDE1540
SVDE1550
SVDE1560
SVDE1570
SVDE1580
SVDE1590

```



```

      W(1) = M
      W(2) = DEL
      W(3) = NH
C
      RETURN
      END
      SUBROUTINE XINT(XX,YY,X,Y)
      DIMENSION XX(15),YY(15)
      I=1
      5 IF(X.LT.XX(I+1))GO TO 10
      I=I+1
      GO TO 5
      10 Y=YY(I)+((X-XX(I))/(XX(I+1)-XX(I)))*(YY(I+1)-YY(I))
      RETURN
      END
0000000000000000000000000000

```

SVDE1600  
SVDE1610  
SVDE1620  
SVDE1630  
SVDE1640  
SVDE1650

# DISTRIBUTION

	No. of Copies
Defense Documentation Center Cameron Station Alexandria, Virginia 22314	6
Commander US Army Materiel Development and Readiness Command ATTN: DRCRD	1
DRCDL	1
5001 Eisenhower Avenue Alexandria, Virginia 22333	
DRCFM-RSE, Mr. Sullivan	1
Mr. Oldham	1
DRSMI-R, Dr. McDaniel	1
Dr. Kobler	1
-LP, Mr. Voigt	1
-FR, Mr. Strickland	1
-RF, Mrs. Stamps	5
-RBD	3
-RPR (Record Set)	1
(Reference Copy)	1